

## **TITLE : DOOR LOCK STATUS MONITORING AND ALERT SYTEM**

----- Prantik Roy

---

**INTRODUCTION** : The Internet of Things (IoT) is revolutionizing how we interact with the physical world by embedding connectivity in everyday objects. In this project, we focus on a "Door Lock Status Monitoring and Alert System," an IoT application designed to enhance home and office security. This system allows users to monitor the status of their door locks remotely and receive real-time alerts if any unauthorized access is detected. By integrating sensors and connectivity, this solution provides peace of mind, enabling users to check if doors are locked or unlocked through a mobile application or web interface.

### **COMPONENTS USED :**

1. RASPBERRY PI PICO - 1 piece
2. Bread Board - 1 piece
3. LED - 4 pieces
4. Buzzer - 1 piece
5. MC-38 Wired Door Window sensor
6. M-M, M-F, F-F
7. Header

### **SOFTWARE OR APPLICATION USED :**

THONNY : Thonny is a user-friendly integrated development environment (IDE) designed with beginners in mind, making it an excellent choice for those new to Python programming. Its clean and simple interface allows users to focus on learning without being overwhelmed by complex features. Thonny offers step-by-step code execution, which helps beginners understand the flow of their programs and easily identify errors. The IDE includes a variable explorer that displays the current state of all variables, aiding in debugging and comprehension. With an integrated Python shell, Thonny allows users to test and experiment with code interactively. It also features automatic syntax error

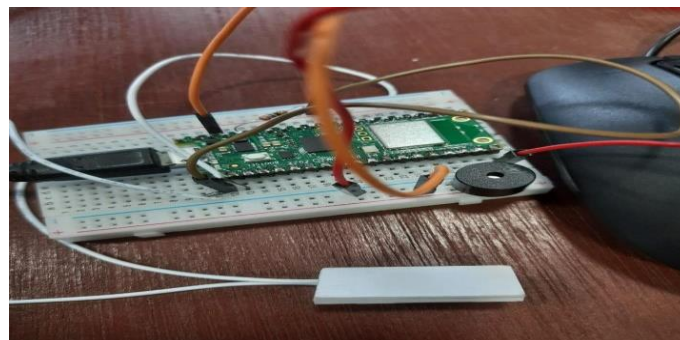
highlighting, a built-in debugger, and a straightforward package manager. Additionally, Thonny supports MicroPython, making it a versatile tool for projects involving microcontrollers like the Raspberry Pi Pico. Overall, Thonny is an ideal IDE for beginners, educators, and hobbyists who seek a simple yet powerful platform to learn and develop Python code.

## MC-38 WIRED DOOR WINDOW SENSOR :

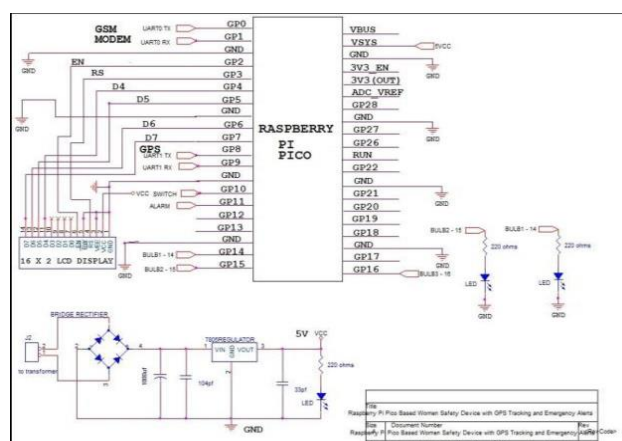
The MC-38 wired door/window sensor is a commonly used magnetic contact sensor designed for security systems. It consists of two main components: a magnet and a reed switch. The reed switch is typically mounted on the stationary part of a door or window frame, while the magnet is mounted on the moving part of the door or window itself. When the door or window is closed, the magnet is in close proximity to the reed switch, causing it to close the circuit. If the door or window is opened, the magnet moves away, causing the circuit to open and triggering an alert or alarm in the connected security system.

This type of sensor is often used in residential and commercial security setups to monitor entry points. The MC-38 is favored for its reliability, ease of installation, and compatibility with various wired alarm systems. It provides a simple yet effective method for detecting unauthorized entry, enhancing the overall security of a property.

## CIRCUIT IMPLEMENTATION THE IOT:



## ARCHITECTURE :



## CODE :

```
import machine
import time
from machine import Pin
from time import sleep
import urequests as rq
lockstat=True
unlockstat=True
TELEGRAM_BOT_TOKEN ='7503505246:AAH9H1QTREtYtXAvRp1b9zljLXfClpjc0N8'
TELEGRAM_BOT_ID = '6119294750'
TELEGRAM_API_URL =
'https://api.telegram.org/bot{}/sendMessage'.format(TELEGRAM_BOT_TOKEN)
def do_connect():
    SSID='prantik'
    PASSWORD='12345678'
    import network
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    if not wlan.isconnected():
        print('connecting to network...')
        #wlan.connect('LAPTOP-0FTK45F2 3896', '8V6387f{')
        wlan.connect(SSID,PASSWORD)
        while not wlan.isconnected():
            pass
        print('network config:',wlan.ifconfig('addr4'))
def send_telegram_message(message):
    payload = {'chat_id':TELEGRAM_BOT_ID,'text':message}
    try:
        rq.post(TELEGRAM_API_URL,json=payload)
        return True
    except Exception as e:
```

```

        print("Error Sending Message",e)
    return False
d=Pin=(0,Pin.IN)
#current_time = time.gmtime()
#current_date = time.localtime()
def dataupload(d):
    api = 'https://door-lock-system-3c238-default-rtdb.firebaseio.com/ .json'
    print('sending data...')
    re = rq.post(api,json=d).text
    print(re)
do_connect()
while True:
    doorstat=d.value()
    print('Status:',doorstat)

    dt = time.localtime()
    y = dt[0]
    mo = dt[1]
    day = dt[2]
    h = dt[3]
    m = dt[4]
    s = dt[5]
    date = f'{day}/{mo}/{y}'
    ti = f'{h}:{m}:{s}'
    dic = {
        'status': doorstat,
        'time':ti,
        'date':date
    }
    sleep(0.2)

```

```

if(doorstat==0
):
    buzzer.value
    ue(0)
    print("Door
    is locked")if
    lockstat ==
    True:
        datauploa
        d(dic)
        lockstat =
        False
        unlockstat
        = True
elif(doorstat == 1):
    print(f"Alert!Door is unlocked at: {date}
    {ti}")buzzer.value(1)
    buzzer.
    on()
    sleep(0
    .3)
    buzzer.v
    alue(0)
    sleep(0
    .3)
    lockstat
    = True
    if unlockstat ==
        True:
            dataupload(

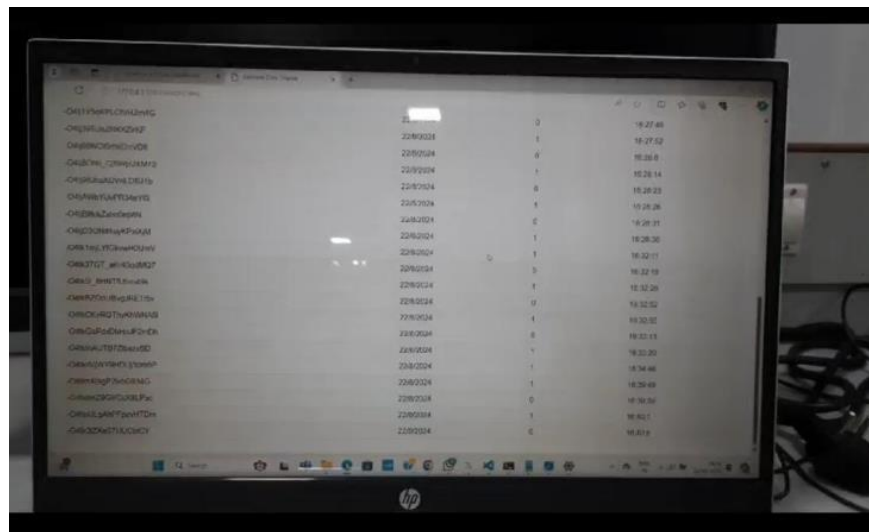
```

```

dic)
send_telegram_message("Door is
unlocked")unlockstat = False
else:
print("no sensor applied")

```

## DATABASE OF THE IOT PROJECT :



ID	Name	Status	Time
1	GATEWAY	1	10:27:00
2	DOOR	1	10:27:02
3	WINDOW	1	10:27:04
4	ALARM	1	10:27:06
5	DOOR	1	10:27:08
6	WINDOW	1	10:27:10
7	ALARM	1	10:27:12
8	DOOR	1	10:27:14
9	WINDOW	1	10:27:16
10	ALARM	1	10:27:18
11	DOOR	1	10:27:20
12	WINDOW	1	10:27:22
13	ALARM	1	10:27:24
14	DOOR	1	10:27:26
15	WINDOW	1	10:27:28

## CONCLUSIONS :

The "Door Lock Status Monitoring and Alert System" is an excellent example of how IoT technology can be harnessed to enhance security and convenience in everyday life. By providing real-time monitoring and alerts, this system not only ensures the safety of homes and offices but also offers users greater control and peace of mind. As IoT continues to evolve, its applications will expand further, integrating more deeply into various aspects of daily life and industry, paving the way for smarter, more efficient systems.